

UNIVERSIDAD POLITÉCNICA SALESIANA SEDE QUITO

CARRERA: INGENIERÍA DE SISTEMAS

**Trabajo de titulación previo a la obtención del título de:
INGENIERA E INGENIERO DE SISTEMAS**

**TEMA:
BASES DE DATOS NO RELACIONALES: UTILIZACIÓN DE MONGO DB COMO
BASE DE DATOS NO RELACIONAL EMPLEANDO FORMATO GEO JSON**

**AUTORES:
DAYSI YADIRA CHANGO GAVILÁNEZ.
JONATAN JAVIER JÁCOME SALAS.**

**TUTOR:
ING. GUSTAVO ERNESTO NAVAS RUILOVA**

Quito, julio de 2016

CESIÓN DE DERECHOS DE AUTOR

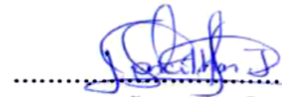
Nosotros, Daysi Yadira Chango Gavilánez con documento de identificación N° 1724471691 y Jonatan Javier Jácome Salas con documento de identificación N° 1720588498, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de grado intitulado: BASES DE DATOS NO RELACIONALES: UTILIZACION DE MONGO DB COMO BASE DE DATOS NO RELACIONAL EMPLEANDO FORMATO GEO JSON, mismo que ha sido desarrollado para optar por el título de: Ingeniera e Ingeniero de Sistemas en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.



Daysi Yadira
Chango Gavilánez

CI: 1724471691



Jonatan Javier
Jácome Salas

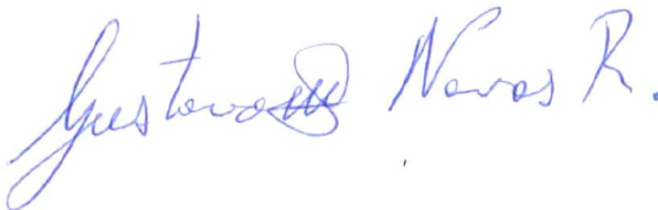
CI: 1720588498

Fecha: julio de 2016.

DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR

Yo, declaro que bajo mi dirección y asesoría fue desarrollado el trabajo de titulación BASES DE DATOS NO RELACIONALES: UTILIZACION DE MONGO DB COMO BASE DE DATOS NO RELACIONAL EMPLEANDO FORMATO GEO JSON realizado por Daysi Yadira Chango Gavilánez y Jonatan Javier Jácome Salas, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana para ser considerados como trabajo final de titulación.

Quito, julio de 2016.



.....
GUSTAVO ERNESTO NAVAS RUILOVA
CI: 1705675625

Bases de Datos No Relacionales: Utilización de MongoDB como Base de Datos No Relacional Empleando Formato Geojson

Daysi Chango Gaviláñez, Jonatan Jácome Salas, Gustavo Navas Ruilova

Abstract—The result of this job was the re-engineering of a Geo Portal, which already exists, to the use of a non-relational database, which is managed with MongoDB, whose formats are on JSON. The information that exist in the Geo portal is about the work that every Salesian house represents. As result, it makes it easy to associate all the activities that every house is involved to. Currently, ten houses are been used out of the 24 for this project. The use of a depth-advanced searching to log on the data, is the most characteristic feature that MongoDB owns. This facilitates the loading of data into the menus and sample results. Throughout the development, it was observed that loading data was effortless because the similarity that is shared between the internal structure for the Salesian houses and JSON format, as well the quick access to specific information with specific queries that owns the MongoDB.

Index Terms— Depth searching, geo portal, Gis, json, MongoDB, non-relational database, NoSQL.

Resumen—Este trabajo es el resultado de la re ingeniería de un Geo Portal ya existente empleando estructuras de datos no relacionales (NoSQL) como MongoDB que permite manipular datos en formato JSON. El Geo portal contiene información acerca de casas Salesianas que se asocian con obras, de tal manera que una casa salesiana puede tener una o más obras. Actualmente hay 24 casas, para este trabajo se tomaron 10 de ellas. Se utilizaron búsquedas avanzadas o búsquedas en profundidad para el acceso a los datos, estas constituyen una de las características más importantes de MongoDB puesto que facilitan la carga de información en los menús y la muestra de resultados.

Durante el desarrollo se pudo observar que la carga de datos fue más sencilla debido a la similitud entre la estructura interna de la comunidad salesiana y el formato json.

De la misma manera la localización de información con sentencias propias de MongoDB.

Palabras Clave— Bases de datos no relacionales, búsquedas en profundidad, geo portal, gis, json, MongoDB, NoSQL.

1. INTRODUCCIÓN

Los sistemas de información geográfica cada vez son más utilizados en nuestra vida cotidiana, puesto que desempeñan un papel importante en el transporte, la planificación y gestión de recursos, y a su vez constituyen una de las herramientas más potentes para el manejo de la información. [1]

Poco a poco nace la necesidad de almacenamiento de datos en grandes cantidades, para esto se ha utilizado sistemas gestores de base de datos libres como MongoDB. [2]

Artículo enviado para revisión al 30 de Junio del 2016. El material investigativo empleado en la realización del mismo fue proporcionado por parte de la Inspección Salesiana.

Daysi Chango Gaviláñez, Egresado de Ingeniería de Sistemas - Universidad Politécnica Salesiana - Quito. Autor para correspondencia ✉: dchango@est.ups.edu.ec

Jonatan Jácome Salas, Egresado de Ingeniería de Sistemas - Universidad Politécnica Salesiana - Quito. (E-mail: jjacomes@est.ups.edu.ec).

Gustavo Navas Ruilova, Grupo de Investigación IDEIAGEOCA - Universidad Politécnica Salesiana, Docente. (E-mail: gnavas@ups.edu.ec)

A continuación se puntualiza algunos conceptos utilizados en nuestro trabajo.

MongoDB es una base de datos No SQL o no relacional que permite guardar los datos en documentos que son manejados a través de colecciones. Estos documentos son de tipo json por esta razón manejan un formato liviano y fácil de interpretar tanto para máquinas como para humanos. [3]

Las bases de datos NoSQL no siguen un esquema, es decir que los objetos de una colección trabajan bajo un criterio parecido a una tabla de una base de datos relacional, y esta tabla puede tener distintos tipos de datos. [4]

Las bases de datos relacionales satisfacen las características de Consistencia y disponibilidad denominadas CA, pero tiene serios problemas con la Tolerancia a muchos nodos, para la nube se requiere escalabilidad y se necesita sacrificar consistencia. [5]

Las Bases de Datos no relacionales, conocidas también como bases de datos documentales, en cambio tratan un concepto similar al ACID y se denomina para ellas BASE (Basically Available, Soft-State y Eventual Consistency) donde se prioriza la disponibilidad sobre la consistencia, es decir que el sistema probablemente no será consistente en cada instante del tiempo disponible. [5] [6]

Utilizando los parámetros anteriores se implementó el geo portal en su primera versión, con las herramientas de Geo referenciación, una base de datos no relacional llamada MongoDB, una base de datos PostgreSQL y el entorno de desarrollo JSF más el Framework Prime Faces propio de Java.

2. MATERIALES Y MÉTODOS

2.1. Fase de revisión

Actualmente existen 24 casas salesianas, 3 novicios y 155 hermanos salesianos en todo el Ecuador. [7]

Las casas Salesianas se asocian con las obras salesianas de tal manera que una casa salesiana puede tener una o más obras Salesianas que pueden ser de tipo: Social, Parroquial, Escolar, Salud, Medio de Comunicación, Auto gestionada u otros. [7]

Para el desarrollo de nuestro trabajo hemos tomado una muestra de 10 casas salesianas y nos basaremos en la estructura interna de la comunidad para la muestra de información en nuestro Portal. Se presenta en la Figura 1 la estructura interna de la comunidad salesiana.

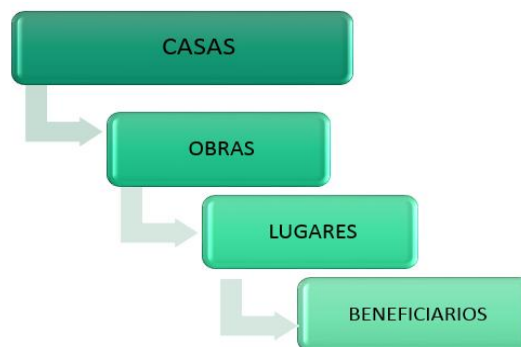


Figura 1. Estructura interna de la comunidad salesiana.

Es importante mencionar que el Portal para la comunidad Salesiana posee versionamientos, para el desarrollo de nuestro trabajo tomaremos como base el proyecto de titulación (Análisis, diseño, construcción e implementación de un geo portal con las herramientas MongoDB, json para el proyecto ide & ups).

2.2. Estructura Anterior

La Primera versión de este sistema poseía una base de datos relacional PostgreSQL, se usaba esta herramienta para almacenar las casas salesianas asignándoles un Id (Identificador único) que permitía buscar en los archivos json manejados por MongoDB, en donde se almacenaba la información de cada una de las casas salesianas, (Código, Nombre, Denominación, Pais, Región, Provincia, Cantón, Parroquia, Sector, Responsable, Dirección, Telefono, horario, Email, Tipo, Campo, Nro Personas, Beneficiario, etc) en conclusión usaba una estructura mixta es decir que combinaban el almacenamiento de datos en bases de diferentes tipos.

En la Figura 2 se muestra de manera gráfica la estructura que manejaba inicialmente el portal.

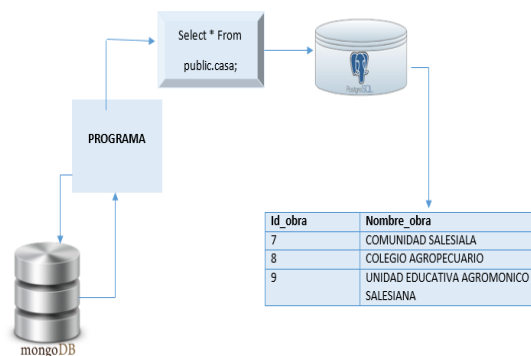


Figura 2. Estructura de datos antigua del portal.

2.3. Comparativa de búsquedas entre PostgreSQL vs MongoDB

Como fase inicial vamos a establecer una comparativa de búsquedas entre PostgreSQL y MongoDB, para visualizar las diferencias de cada una, en una sentencia básica. La siguiente sentencia busca todas las casas que se encuentra activas o en servicio.

PostgreSQL

```
SELECT id_lug, estado
FROM casas
WHERE estado = 'A'
```

Esta sentencia permite buscar las casas que se encuentran en servicio.

```
SELECT *
FROM casas
WHERE estado != 'A'
```

A diferencia de esta sentencia que permite buscar las casas que no se encuentran en servicio.

MongoDB

```
db.getCollection('casas').find(
  { estado: "A" })
```

```
db.getCollection('casas').find(
  { estado: { $ne: "A" } })
```

```
SELECT *
FROM casas
WHERE provincia =
'PICHINCHA'
```

```
db.getCollection('casas').find(
  { provincia: "A" })
```

Ahora ejecutamos la sentencia que nos permite buscar todas las casas que se encuentran en la provincia de Pichincha.

2.4. Estructura Actual

La estructura actual eliminó el uso de PostgreSQL y se dispuso a trabajar únicamente con MongoDB que es capaz de manejar la misma cantidad de datos estructurados en colecciones, los archivos que manejan estas colecciones son de tipo json es decir formato plano, no utilizan muchos recursos para ninguna de las actividades en las que se los usa, por esta razón permite optimizar el tiempo de búsqueda en el portal.

Después de realizar varios cambios llegamos a la nueva estructura que se puede apreciar en la Figura 3.

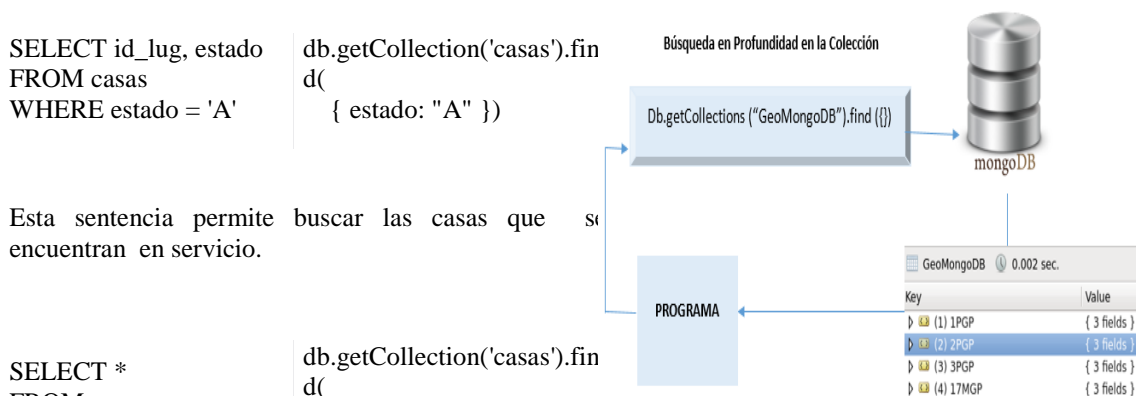


Figura 3. Estructura de datos nueva del sistema.

2.5. Búsqueda en profundidad en el Geo portal

Las búsquedas en MongoDB se ejecutan con javascript.[9]

Una manera de acceder a cualquier nivel de datos en la colección que se localiza en nuestro trabajo, es mediante la función find, la cual se busca documentos con o sin parámetros dentro de las colecciones, se manejan bajo una estructura

seguida de un punto que separa cada parámetro por ejemplo:

En esta colección accedemos a tres niveles de profundidad, como se muestra en la Figura 4, cada nivel se encuentra separado por un punto seguido del valor del atributo, como se puede observar en el siguiente extracto de código, donde obtenemos todas las casas que se encuentran en la provincia del Azuay.

```
“db.getCollection('GeoMongoDB').find({"features.
properties.Provincia":"AZUAY"})”
```

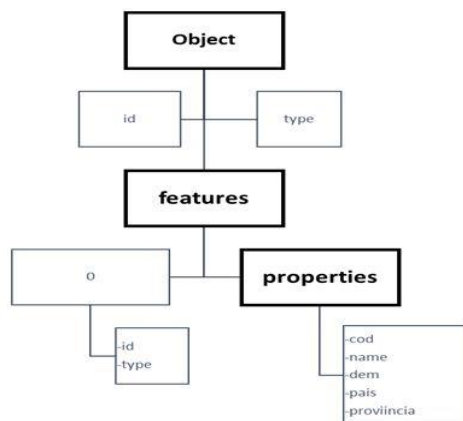


Figura 4. Modelo lógico para búsquedas.

La Figura 5 muestra dicha consulta en el nivel 1 de la colección de datos para el efecto se ha utilizado la herramienta Robomongo.

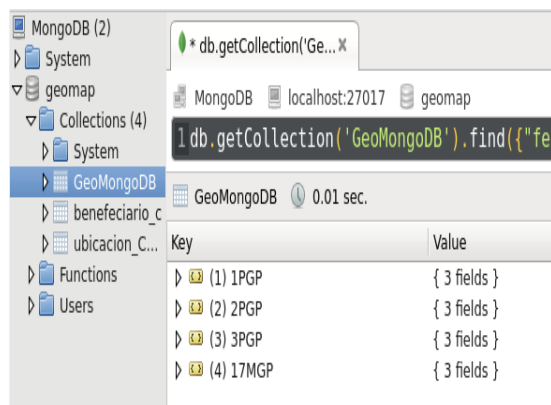


Figura 5. Nivel uno de colección de datos.

De la misma manera la Figura 6 muestra el nivel 3 de la colección de datos.

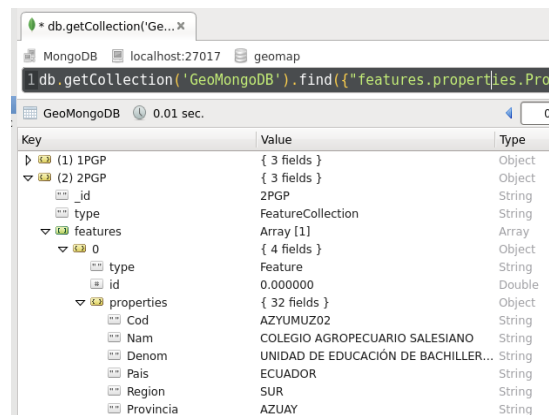


Figura 6. Niveles de datos en colecciones.

En la Figura 7 se puede apreciar el resultado de la siguiente consulta:

```
db.getCollection('GeoMongoDB').find({"_id":"2PGP"})
```

En esta consulta se usa como criterio de búsqueda un identificador, que puede ser de dos tipos: el PGP que hace referencia a un punto geográfico de acuerdo a coordenadas fijas y el MGP que hace referencia a un multi polígono geográfico de acuerdo al área de influencia.

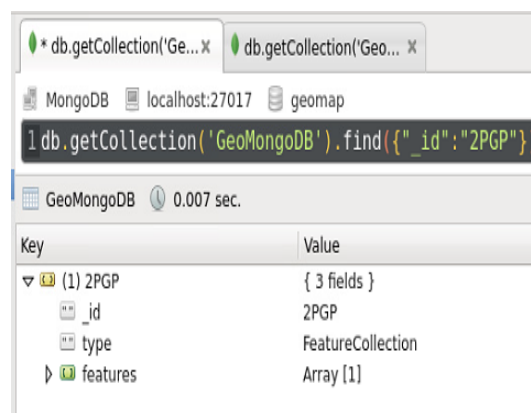


Figura 7. Búsqueda en mongodb determinando un atributo.

2.6. Consulta con tipo de dato lógico o booleano.

Al realizar la siguiente consulta podemos discriminar solo los objetos que contienen los atributos features.

```
db.getCollection('GeoMongoDB').find({"Collection.features: { $exists: true } })
```


Y al realizar la consulta con valor falso, nos muestra solo los objetos que no contienen atributos features.

```
db.getCollection('GeoMongoDB').find({"Collection.features": { $exists: false } } ).
```

Estas consultas permiten obtener valores, de acuerdo a un dato booleano.

2.7. Comparativa entre PostgreSQL y MongoDB empleando java

A continuación se muestra un extracto de código que permite visualizar el modo de consulta en PostgreSQL usando java.

2.7.1 Modo de Consulta en PostgreSQL usando JAVA

Se consulta todas las casas en una tabla llamada Casa

```
public Casa buscarCasaPorId(Integer id){
    System.out.println("ID CASA A BUSCAR "+id);
    Query consulta = em.createQuery("select c
from Casa c where c.id = :id");
    consulta.setParameter("id", id);
    return(Casa) consulta.getSingleResult();
}
```

2.7.2 Modo de Consulta en MongoDB usando JAVA

Ahora para obtener los datos de una colección o lista debemos identificar a un objeto con un atributo en este caso será el Id o identificación para saber que es una casa (punto geográfico) y no una área de influencia (multi polígono).

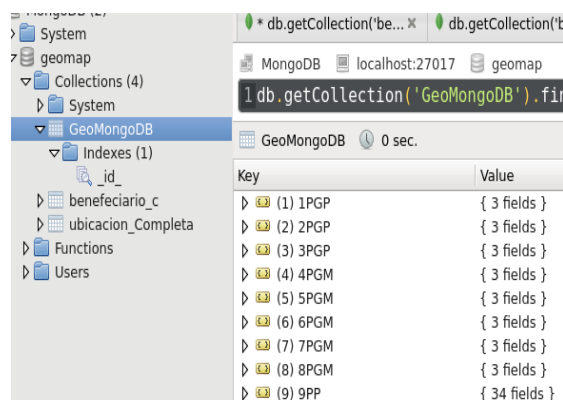
El siguiente extracto de código realiza lo antes mencionado.

```
public ArrayList<SelectItem> Mostrar()
{
    Conexion conec = new Conexion();
    DBCursor cursor =
conec.consultarColeccionporParametros("Casa", "$exists", true);
    String casa = "";
    while (cursor.hasNext())
    {
```

```
DBObject obj = cursor.next();
if
(!obj.get("Casa").toString().trim().equalsIgnoreCase(casa.trim()))
{
    NombresCasas.add(new
SelectItem(String.valueOf(obj.get("Casa").toString()),obj.get("Casa").toString()));
}
NombresCasas.get(i).getValue();
//i++;
}
prov=obj.get("Provincia").toString();
}
return NombresCasas;
}
```

Los métodos de consultas expuestos anteriormente se usaron para búsquedas de colecciones de datos en MongoDB y de esta manera descartar completamente la base de datos PostgreSQL.

De la misma manera realizamos la adaptación de dichos métodos de consulta para mostrar los nuevos datos en el portal. En la Figura 8 se puede apreciar los resultados de dicha consulta.



Key	Value
(1) 1PGP	{ 3 fields }
(2) 2PGP	{ 3 fields }
(3) 3PGP	{ 3 fields }
(4) 4PGM	{ 3 fields }
(5) 5PGM	{ 3 fields }
(6) 6PGM	{ 3 fields }
(7) 7PGM	{ 3 fields }
(8) 8PGM	{ 3 fields }
(9) 9PP	{ 34 fields }

Figura 8. Método de consulta en mongodb.

Las búsquedas de datos en el sistema se realizan por Id_lugar para que establecer una relación entre dos colecciones en este caso colección ubicación completa y Beneficiario de esta manera se delimita el área de influencia.

Este extracto de código permite visualizar el área de influencia, que pertenece a la casa que se elija.

```
public void mapaMultiPoligono() {
    cursor =
con2.consultarColeccionPorcualquiermpara("id_lug
",id_lug);
    System.out.println("ID DE LUGAR
"+id_lug);
```



```
String coordenadas = "";
while (cursor.hasNext())
{
    DBOObject obj = cursor.next();
    String loc = obj.get("geometry").toString();

    coordenadas=loc.substring(loc.indexOf("\coordina
ates\\")+19,loc.indexOf("]]}"));
}
mapaPoligono(coordenadas);
}
```

3. Implementación del Geo portal

Se realiza la configuración bajo una arquitectura Linux, en donde se aloja el portal y sus datos correspondientes.

Las Figuras 9, 10, 11 son claros ejemplos de como los datos que se encuentran almacenados en MongoDB, se visualiza tanto las obras salesianas (puntos geográficos) como sus beneficiarios (área de influencia).

El acceso público al sistema se encuentra en la siguiente página web:

<http://190.15.136.4:8090/Mongo2/Home.jsf>

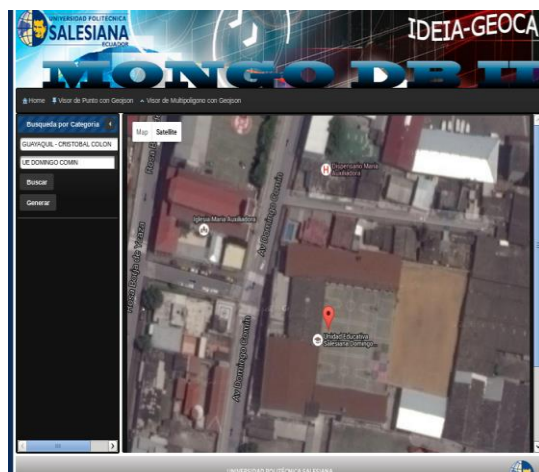


Figura 9. Casa salesiana.

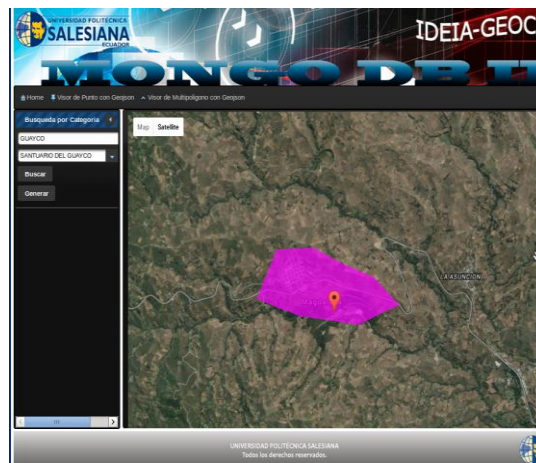


Figura 10. Casa salesiana y área de influencia.

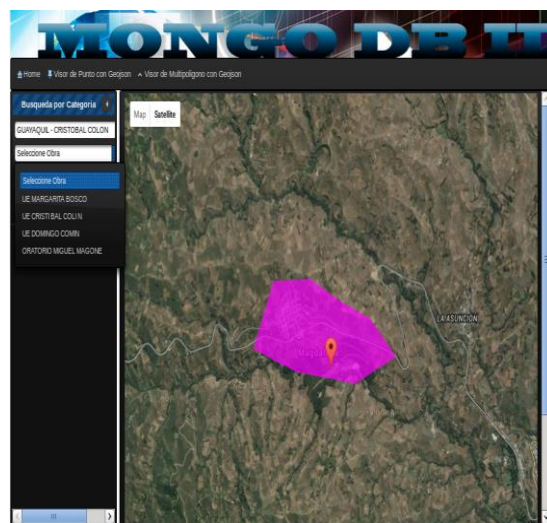


Figura 11. Casa salesiana y área de influencia.

4. Conclusiones

Se investigó que las prioridades del modelo relacional se basan en los datos, operaciones, gestión y su manipulación, mientras que en bases de datos no relacionales se prioriza la optimización, rendimiento y escalabilidad [5]

Se determinó que cada base de datos ya sea SQL o MongoDB gestiona sus ventajas y desventajas sobre algún contenido en particular, por tal razón es importante establecer la prioridad de solución del problema.

Al referirse a consultas: SQL puede extraer los datos siguiendo el camino de un árbol jerárquico, ya sea desde el nodo superior hacia los inferiores o viceversa, mientras que en la estructura NOSQL

permite desplazarse de manera jerárquica únicamente desde el nodo superior hacia el inferior.

Se establece una similitud sobre la estructura jerárquica que maneja la comunidad salesiana vs el formato json que contiene la información salesiana es decir que una casa salesiana puede tener una o más obras Salesianas que pueden ser de distintos tipos y operar un área de influencia de acuerdo al caso.

La distribución de datos en MongoDB, facilita la carga de información en los menús para la muestra de resultados.

En MongoDB, las búsquedas avanzadas o búsquedas en profundidad constituyen una de las característica más importantes, determinadas por su sencillez y la obtención de cualquier tipo de dato a nivel de arreglos.

Cabe mencionar que MongoDB es una herramienta que resuelve problemas de desempeño, almacenamiento y procesamiento masivo. Es importante tener conocimiento de este tipo de bases de datos puesto que nos encontramos en un mundo en el que se generan millones de datos por segundo, con esta herramientas obtendríamos soluciones sencillas para grandes problemas.

Agradecimientos

El presente trabajo de investigación fue realizado gracias al amplio material investigativo que posee la Inspectoría Salesiana, a quienes nos gustaría extender nuestro más profundo agradecimiento, por entregarnos la información necesaria para el desarrollo de este trabajo en particular.

Referencias

- [1] Li, Jie, Zhuo Liu, Chunxiao Xing, Yong Zhang, and Chao Li. "Research and Implementation of the Document-Oriented Database Test Methods for GIS Applications," 3–6. IEEE, 2013. doi:10.1109/WISA.2013.8.
- [2] Huang, Qunying, and Chen Xu. "A Data-Driven Framework for Archiving and Exploring Social Media Data." *Annals of GIS* 20, no. 4 (October 2, 2014): 265–77. doi:10.1080/19475683.2014.942697.
- [3] Rubén Fernandez. "MongoDB: Qué Es, Cómo Funciona Y Cuándo Podemos Usarlo (o No)." *Genbeta Dev*, February 3, 2014. <http://www.genbetadev.com/bases-de-datos/mongodb-que-es-como-funciona-y-cuando-podemos-usarlo-o-no>.
- [4] Vergara, Alida. "Mongo DB, Rápido Y Escalable | Tech Blog for Developers | Facilcloud." Accessed February 12, 2016. http://www.facilcloud.com/noticias/?p=2500&lang=es_ES.
- [5] William Díaz Sepúlveda. "Bases de Datos NOSQL." *BASES DE DATOS NOSQL: LLEGARON PARA QUEDARSE*, May 27, 2013. <http://basededatosnosql.blogspot.com/>.
- [6] Yohan Graterol. "NoSQL Como El Futuro de Las Bases de Datos," March 10, 2014. <http://www.maestrosdelweb.com/nosql-como-el-futuro-de-las-bases-de-datos/>.
- [7] "Salesianos Ecuador." Presentación. Accessed February 1, 2016. <http://www.salesianos.org.ec/quienesp/1/>.
- [8] Malaya, Victoria. "SQL vs. NoSQL: Indexes Comparison between MongoDB and MS SQL Server.," November 8, 2013. <http://sql-vs-nosql.blogspot.com/2013/11/indexes-comparison-mongodb-vs-mssqlserver.html>.
- [9] Rubén Fernandez. "MongoDB: Encuentra Tus Datos Con Consultas Simples." *Genbeta Dev*, October 17, 2014. <http://www.genbetadev.com/trabajar-como-desarrollador/mongodb-encuentra-tus-datos-con-consultas-simples>.